Algorithm: Lab14 (By Sujiv Shrestha ID:610145)
Problem 1.

1.Show that TSP is NP-complete. (Hint: use the relationship between TSP and HamiltonianCycle discussed in the slides. You may assume that the HamiltonianCycle problem is NP-complete.)

Solution:

Let us consider we have Graph G=(V,E) which is subgraph with vertices V_n. Now we can show that HamiltonianCycle problem with this graph as input is reducible to TSP inorder to show that TSP is NP-complete as we know that HamiltonianCycle problem is NP-complete as well.

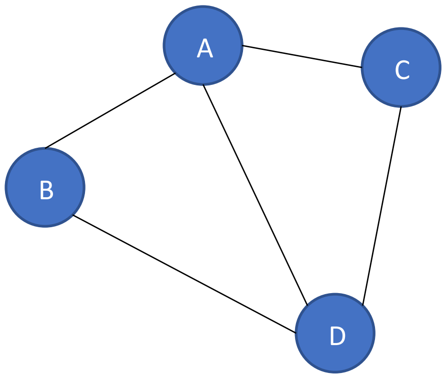


Figure: Example of G

For this let us obtain an instance H, w, k of TSP where H is complete graph with same n vertices (V_n) as in G, obtained by adding missing edge in G such that,

$$w(e) = \begin{cases} 0, & e \in E \\ 1, & e \notin E \end{cases}$$

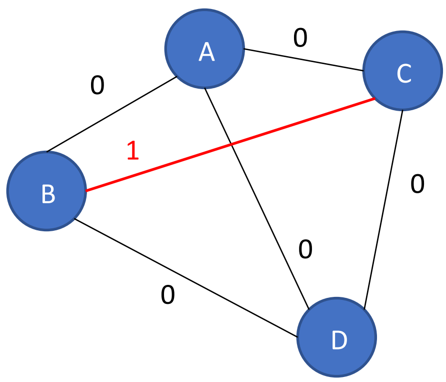


Figure: Example of H obtained from G

Let k=0,

So, to show that G and H has same solution we need to show that G has HamiltonianCycle iff H has a Hamiltonian cycle with sum of the edge weight/cost ≤ k.

Let, C be the HamiltonianCycle in G then C is Hamiltonian Cycle in H as well since G is subgraph of H. And since all the edge in H that is also in G have weight 0 we can say that sum of the edge weight in C of H is less or equal to k (i.e, 0). Hence C is also Hamiltonian Cycle in H.

Problem 2.

2.Below is another variation of the Knapsack problem.Given a set S= {s0, s1, ..., sn-1} of items, weights {w0, w1,..., wn-1}, values {v0, v1, ..., vn-1},a max weight W, and a min value V, find a subset T of S whose total value is no less than V and total weight is at most W.Show that the SubsetSum problem is polynomial reducible to this Knapsack problem.

Solution:

Given: Set of items S= {s0, s1, ..., sn-1},
weights w= {w0, w1,..., wn-1} and
values v= {v0, v1, ..., vn-1}

So, if T is the solution of this Knapsack problem then following must satisfy:

$\sum_T w_i \leq$ W (max. weight)

and also,

$\sum_T v_i \geq$ V (min. value)

To show that SubsetSum problem is polinomial reducible to knapsack problem, let us consider SubsetSum problem with set of item S= {s0, s1, ..., sn-1} and values V= {v0, v1, ..., vn-1} and the required sum be M.

Now, lets us consider a knapsack problem with same items as in SubsetSum problem with same values V and the weight w = v = {v0, v1, ..., vn-1}. Let the maximum allowed weight W = M and minimum required value V = M.

Now we need to show that T is solution for SubsetSum problem iff it is also the solution for knapsack problem. For T to be the solution of knapsack problem above following must be true.

$\sum_T v_i \geq$ M Equation i

and also,

$\sum_T w_i \leq$ M Equation ii

Since, $w_i = v_i$, we have,

$\sum_T v_i \leq$ M

The equation i and equation ii are true iff $\sum_T v_i =$ M. Which satisfies that T is also the solution for SubsetSum problem.

Problem3.

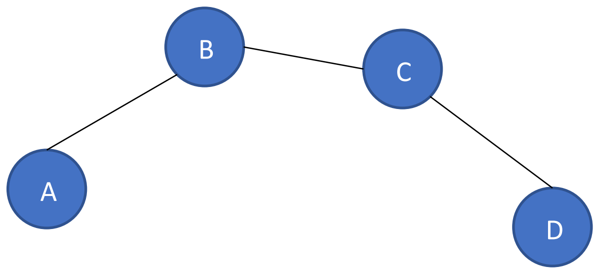
3.Show that the worst case for VertexCoverApprox can happen by giving an example of a graph G which has these properties:

a.G has a smallest vertex cover of size s

b.VertexCoverApprox outputs size 2*s as its approximation to optimal size.

Solution:

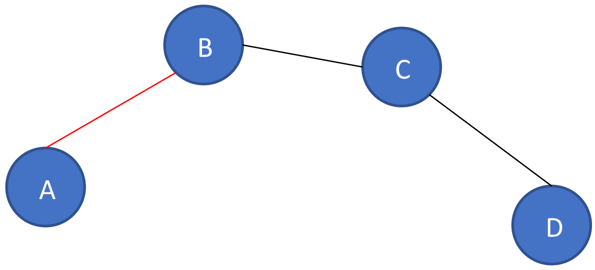
Lets consider following graph G=(V,E) which has a smallest vertex cover of size 2



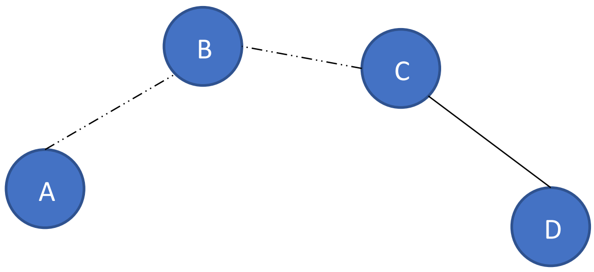
The smallest vertex cover C` of this graph is of size 2. For eg. C` = {A,C} or {B,C} or {B,D} etc.

Now using VertexCoverApprox to find the vertex cover of G

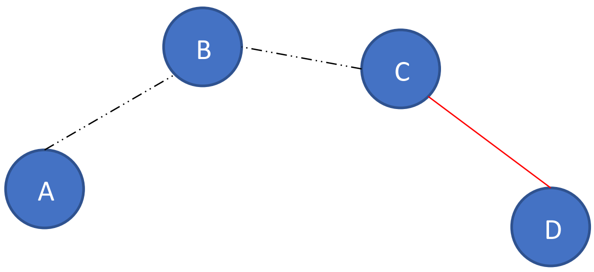
Step 1: C←{}
Step 2:



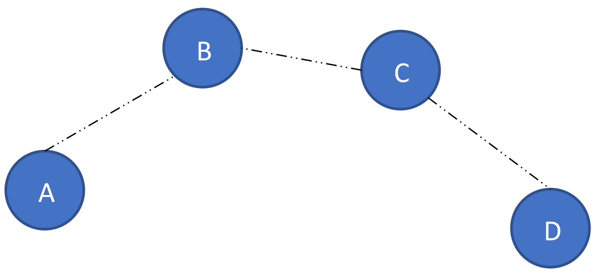
C←{A,B}
Remove edges that is incident to A and B



Step 3:



C←{A,B,C,D}
Remove edges that is incident to C and D



Now, we have |C| = 2*|C`|

To generalize this let us consider a graph G=(V,E) with n vertices and m edges. Let VetexCoverApprox runs over r edges out of m edges. Now from the definition of vertex cover we know that the vertex cover of the graph has size ≥ r.

|Smallest vertexCover| = r

But since, the VertexCoverApprox adds both vertex of the edges it runs over, the size of vertex cover from this algorithm is 2*r

|VertexCoverApprox. VertexCover| = 2*r

So, we have |VertexCoverApprox. VertexCover| = 2*|Smallest vertexCover|

Problem4.

4.The decision problem formulation of the Vertex Cover problem is this: Given a positive integer k, and a graph G, is there a vertex cover for G having size ≤ k? Show that this decision problem belongs to NP.

Solution:

The decision problem given in the question can be broken down into two parts.

Part1: Check if the given set of vertices is a VertexCover problem of graph G or not.

Part2: Check if the size of the given set of vertices ≤ k

We know that the decision problem in part1 belongs to NP and the part2 as well which runs in O(n). So this shows that the given decision problem belongs to NP.

Alternatively,

Let, us consider the given formulation of the Vertex Cover problem with value of k = n (no. of total vertices in the graph). Since it is obvious that any vertex cover of a graph will have size less than or equal to the no. of vertices in the graph, this will give common solution to the VertexCover problem and the formulation of VertexCover problem given in the question.

This means, VertexCover problem is reducible to the given formulation of VertexCover decision problem. And as we know VertexCover problem belongs to NP-complete, the given decision problem also belongs to NP-Complete and hence beongs to NP.